

A Guide to the Implementation of the National Electronic Disease Surveillance System (NEDSS) in State Public Health Agencies



Council of State and Territorial Epidemiologists

Edited by: Gianfranco Pezzino, MD, MPH

A Guide to the Implementation of the National Electronic Disease Surveillance System (NEDSS) in State Public Health Agencies



Council of State and Territorial Epidemiologists

Edited by: Gianfranco Pezzino, MD, MPH
State Epidemiologist, Kansas Department
of Health and Environment

Table of Contents

Executive Summary.....	ii
Acknowledgments	iii
 I ~ Objectives	 1
Target audience.....	1
Goals.....	1
 II ~ NEDSS and the NEDSS architecture.....	 1
Background	1
The concept of “integrated system”: the integration continuum	2
The multiple layers of modern information systems.....	2
The NEDSS architecture and its elements	4
 III ~ NEDSS implementation options for state programs	 7
Fully integrated system	7
Separate system sharing common tools (“data warehouse”).....	9
CDC-developed NEDSS software	14
State-developed applications.....	15
How many systems should be targeted for inclusion in NEDSS?.....	15
How to choose the best model.....	16
 IV ~ Fitting NEDSS into a state plan: resources and other issues.....	 17
NEDSS requires state resources.....	17
The cost of NOT implementing NEDSS.....	18
Connectivity and security issues	18
Electronic data exchange.....	19
Legal and ethical issues.....	21
 FAQs	 23
Glossary of Terms and list of Acronyms	25



Executive Summary

This document is targeted to program managers and surveillance staff in state health agencies who are involved in the implementation of the National Electronic Disease Surveillance System (NEDSS). The target audience includes epidemiologists and other professional staff with varying degrees of computer knowledge and skills. The goals of the document are:

- A ~ To present the basic principles of the NEDSS architecture and how they affect state-based surveillance systems;*
- B ~ To assist state surveillance programs in deciding how to implement the NEDSS architecture;*
- C ~ To discuss some current issues related to the NEDSS implementation.*

The document is divided into three sections. Section I describes NEDSS and the NEDSS architecture. Included in that section is a discussion of what an integrated system is and how the integration of multiple systems can be achieved in different ways. The multiple layers of modern information systems (i.e., user interface, middle layer with business rules, and database) are also discussed. Finally, the NEDSS architecture and its elements are summarized.

Section II deals more in detail with NEDSS implementation options for state programs. Fully integrated systems are compared with data warehouses. The option of using CDC-developed NEDSS software is discussed against the alternative of adopting state-developed applications. Some guidance is provided on how to choose the best solution for each state and how many systems should be included in NEDSS.

Section III includes a brief description of the resources needed at the state level for the implementation of NEDSS. In addition, connectivity, security, and other important issues (such as confidentiality and legal authority to collect and access surveillance information) are discussed.

A section with Frequently Asked Questions (FAQs) and a glossary complete the document.

NEDSS is an important initiative, originated in part as a result of requests and proposals formulated by state epidemiologists and their professional organization, the Council of State and Territorial Epidemiologists (CSTE). It has the potential to change dramatically the way public health surveillance is conducted in this country. We hope that this document, while not exhaustive or comprehensive, can be helpful in assisting those of us who struggle every day to remain an active part of this process.



Acknowledgments

Many individuals provided crucial contributions to this document. Within CSTE, Donna Knutson, Shah Roohi, and Jonathan Stevens were actively involved in the compilation of the document. In addition, the members of CSTE's Executive Committee provided important feedback to finalize the draft (Jerry Gibson, SC; Dr. Henry Anderson, WI; Richard Hopkins, FL; John Middaugh, AK; Jesse Greenblatt, NH; Mathew Cartter, CT; Suzanne Jenkins, VA; Robert Rolfs, UT; and Steven Macdonald, WA).

Many staff members from CDC also provided important input to this document including Claire Broome, Denise Koo, Joseph Reid, Scott Danos, Dan Jernigan, and others.

Garland Land, MO; Gayle Hansen, KS; Perry Smith and Mike Davisson, NY; and Don Ward, FL; provided the information included in the side text boxes on "lessons learned." Rhoda Nicholas, UT; contributed to the section on the concept of data warehouse.

While the document is the result of a team effort, the ultimate responsibility for any errors or inaccuracies is only mine. Please direct your comments to me at the following address:

Gianfranco Pezzino, MD, MPH
State Epidemiologist
Kansas Department of Health and Environment
900 SW Jackson, Suite 1051
Topeka, KS 66612
Phone (785) 296-6536
Fax (785) 291-3775
gpezzino@kdhe.state.ks.us

April 2, 2001



I ~ Objectives

1) Target audience

This document was prepared by CSTE members and staff, with substantial input from CDC staff from the National Electronic Disease Surveillance System project. The primary target audience includes program managers and surveillance staff in state health agencies who are involved with collecting, processing, and analyzing information in electronic format and who need to learn more about the NEDSS concepts and implementation process. The target audience includes primarily epidemiologists and other professional staff with various degrees of computer knowledge and skills. Staff with more technical functions (e.g., computer specialists and programmers) may find the document helpful to understand the purpose of NEDSS, but this is not meant to be a comprehensive technical guidance document.

2) Goals

- A ~ To outline the basic principles of the NEDSS architecture and how they affect state-based surveillance systems;
- B ~ To assist state surveillance programs in deciding how to implement the NEDSS architecture;
- C ~ To discuss some current issues related to the NEDSS implementation.

II ~ NEDSS and the NEDSS architecture

1) Background

The current collaboration between the Centers for Disease Control and Prevention (CDC) and its public health partners includes the implementation and management of over one hundred surveillance and health information systems. These systems may be part of suggested activities for partners in exchange for funding to deliver vital public health services at the state and local levels. Each center, institute and office at CDC has surveillance needs that are met by the collection of data at the state and local level.

Many of these systems have been in place for several years, and were originally commissioned to detect simple disease and disability conditions. All these systems are administered independently and use non-standardized formats for variable definition and grouping. In addition, these systems are, in general, unable to communicate with one another.

One consequence of this situation is the inability to build a true national disease surveillance system. In the absence of such a system, the identification of bioterrorist events or outbreaks with potentially national impact is very problematic. In addition, the presence of multiple, non-integrated systems leads to an undesirable error rate in the records, an inefficient use of time and labor, a potential for under- or over-reporting, and a duplication of efforts. Many people involved in surveillance at the state and local level have suffered from these inefficiencies. CSTE has carried the message of opposition to the mandate for states to use CDC-designated software as a condition to receiving programmatic funding, and has asked CDC to work towards the adoption of more flexible and integrated information system solutions. Finally, the Health Insurance Portability and Accountability Act (HIPAA), with its requirements for electronic data standardization, has given this problem even more visibility and urgency, while offering unprecedented opportunities for the public and private health sectors to interface electronically with each other.



The National Electronic Disease Surveillance System (NEDSS) is in part a response to these issues. Some of the expected advantages of implementing NEDSS are

- Improvement of the disease reporting system, with more timely and accurate information available for decision making and action.
- Establishment of a national electronic disease surveillance system able to identify more promptly public health threats with national implications.
- Integration of information systems currently separated by differences in data formats and codes, and unable to communicate with each other. This integration will provide the ability to pull, edit, and analyze records from different databases in a combined fashion. It will also give authorized users the ability to compare analyses performed on different databases, since the systems will share variable definitions, codes, and grouping rules.
- Interface with clinical databases (e.g., laboratories, clinics) to make medical information that is valuable for surveillance purposes available in electronic format. This will allow, for example, surveillance programs to receive electronic reports from laboratories or health care providers in a more complete and timely fashion with less burden on the health care sector.
- Encouragement for the public health community to think about a standardized information technology that can be used across multiple surveillance systems, both in infectious and chronic disease.
- Some of these functions may already be available using non-NEDSS compliant systems, but they may require extensive programming, recoding, and resources. NEDSS is expected to allow the use of these functions on a widespread, national basis and with less resources.

2) The concept of “integrated system”: the integration continuum

An integrated surveillance system is one in which records located in different information systems can be easily accessed and combined.

This is a fairly broad and nonspecific definition that can be applied to systems profoundly different in their structure. In fact, the concept of integration can be described as a continuum. On one end of the spectrum are systems so tightly integrated that they can be considered as one system. These systems consist of one software application, with one data entry screen, one set of business rules, and one database. On the other end of the spectrum are systems that are completely independent from each other. These systems may run on different platforms, have different data flow processes, and use different database formats. Such systems would be considered as not integrated. Between these two extremes are many intermediate options with various degrees of integration of one or more of their components.

Modern database management technology makes these intermediate solutions possible and, if well implemented, very effective. For example, information from different databases with similar architecture can be searched and displayed together. Another example is using the same data entry screen for input to different databases.

In summary, there are different ways and degrees to which multiple information systems can be integrated. No one solution is better than the others, and the best solution for each situation needs to be found based on several criteria. This decision process will be discussed below, together with examples of different implementation models for an integrated information system.

3) The multiple layers of modern information systems

A modern database system can be described, using a simplified model, as a multi-layer entity with three major components: a user interface, a middle layer, and a database.



A ~ User interface.

This first layer is what the user sees when, for example, a new record is added to the database. In a multi-layer system, the user interface serves to gather the information that the user wants to send to the system, and to transfer that information to the middle layer for further validation and processing. Modern database management applications allow the creation of user interfaces with relatively little effort. For example, a Microsoft Access data entry screen (i.e., user interface, or layer number one) can be easily created so the user can enter information into a disease-reporting system. If the system is built using a multi-layer approach, this Microsoft Access interface could be used to populate a database (i.e., layer number three) built using a different software application (e.g., IBM DB2), as well as, of course, a Microsoft Access database, as explained more in detail below.

B ~ Middle layer.

This second layer usually contains what is often referred to as business rules,¹ checks and edits, or validation rules. Here resides the bulk of programming code for data entry and validation. For example, this layer could contain code that returns an error message if a user enters a record with a date of birth later than today's date, or that calculates the age of a patient based on the date of birth and the current date. This layer is usually "invisible" to the user, but it is essential for many of the functions that the system performs and to assure the accuracy and integrity of the database.

C ~ Database.

This third layer is where the records are physically located and stored. It usually consists of one or multiple computer files stored by a software application in a proprietary format. Some common database storage formats are Microsoft SQL, Microsoft Access, Oracle, IBM DB2, and there are many others.

D ~ How do these three components interact with each other?

Until a few years ago, a typical database management system included a user interface, validation rules, and a physical database, each using proprietary codes and mixed together in an entity difficult to tease apart. Today's modern applications are usually built with an effort to keep the three components as separate as possible, and to adopt industry standards for each component that make these applications easier to transport to different environments and to integrate with each other.

A browser form screen is an example of a standardized user interface (layer number one). The screen is programmed using standard code that most browsers on the market are able to process, so that what a user sees on the screen is virtually the same regardless of the browser being used (e.g., Microsoft Internet Explorer or Netscape Navigator). A browser interface can be used to process records over the Internet through standardized transmission protocols (such as HTTP, the protocol most commonly used on the World Wide Web). For example, an authorized user can connect through the Internet from any location in the world, using any standard browser application, and perform electronic transactions on a database, such as data entry or queries. A browser can also be used outside of the Internet environment. An Intranet is a local network that uses the same browser and communication protocols as the Internet, but it is not connected to the Internet (that is, is not accessible to Internet users). The flexibility of browsers is so high that many current applications use browser screens as their interface, especially when the applications need to be accessed by multiple users from different workstations.

To continue with our example of how the three components work together, when the user enters a new record into the browser form, a series of validation codes and other procedures located in layer number two process the information submitted. If the record is accepted as a valid record, it is then transferred by this middle layer to the database application in layer number three, where



it becomes part of the physical database. In modern, well-designed database management systems, the same user interface and the same middle layer can be used to populate databases with the same structure but using different industry standards (e.g., Microsoft SQL and Oracle). Similarly, records stored in modern databases with a common architecture but different formats can be accessed and displayed as if they were stored in the same physical file.²

Keeping the three layers well separated facilitates the integration of any of these three components into multiple systems. In practice, this process is not as clear-cut as we are describing it, and moving one layer to a different system still requires some recoding to make that layer “talk” to the other layers in the new system. Nevertheless, this is huge progress from legacy systems that could rarely communicate with each other.

E ~ Why does this matter?

Whether a state is going to develop an integrated software application locally, or adopt software that others have developed, one needs to understand to what extent the application is built using a multi-layer approach, and to what extent the layers are kept separated. This will have important implications, for example, if one wants to adopt one of the three layers (e.g., the data entry screen) for multiple information systems, or to modify or replace one layer (e.g., move records to a new database application), or to combine information from the new system with information coming from other sources.

The concept of a multi-layer system is also essential for the understanding of the differences among alternative NEDSS implementation strategies such as a fully integrated system compared to a data warehouse model.

4) The NEDSS architecture and its elements

NEDSS has been characterized as an “architecture” rather than a true “system.” This architecture includes the definition of several elements that are cornerstones for disease surveillance and reporting in state and local public health departments and in other health care settings. The NEDSS elements are designed to enhance the ability to electronically integrate and link together a wide variety of surveillance activities as well as to facilitate more accurate and timely reporting of disease information.

According to a description provided by CDC staff at the CSTE 2000 annual meeting, the NEDSS architecture has the following characteristics:

- It brings structure, standards and interoperability to the information systems elements of surveillance.
- It allows latitude for states and local health departments to choose specific implementations and fulfill local public health needs.
- It is respectful of local, state, and federal data flow issues.
- It provides a structure around which CDC systems can be integrated.
- It facilitates the ready exchange of comparable data and reports between public health organizations without reprogramming.

There are eight architectural elements in NEDSS, as described in the original CDC program announcement, summarized as follows.

A ~ Conduct and support web browser-based data entry and data management.

This element deals primarily with the user interface (i.e., layer number one) of a multi-layer database management system. It involves developing secure, Web browser-based data entry capacity and management for use inside and outside of health departments. Web data entry can be used for data entry and results reporting for use inside of health departments, between local health departments and state health departments, for reporting from and to other sources (e.g., infection control practitioners, small laboratories), and for case management. There are many advantages in adopting a Web browser-based user interface. One advantage is that no proprietary software needs to be installed on the users' computers, beyond a standard browser. This makes deployment and support of the system cheaper and easier than, for example, installing and supporting a Microsoft Access user interface, which would require the installation of proprietary software on each computer accessing the system. Another advantage is that the programming language used for a Web browser user interface is more standardized than for other commercial applications. Therefore a Web browser data entry screen is expected to work in just about the same way regardless of what browser is used to access that screen, and it is also more "portable" from one database system to another.

B ~ Accept, route and process electronic HL7 messages containing laboratory and clinical content.

This element involves developing the capacity to dynamically accept, import, route to other recipients, and process incoming electronic messages in HL7 format, which uses the LOINC and SNOMED coding standards.³ These messages will come, for example, as reports of test results from local clinical laboratories or emergency departments, from HMOs, from CDC laboratories, or as pertinent information from other public health jurisdictions (e.g., in the setting of multijurisdictional outbreaks).

C ~ Implement an integrated data repository.

An ideal NEDSS system will contain a data repository that will be integrated (i.e., with data from multiple state-based and CDC categorical programs), and patient-centered where appropriate (i.e., where reporting information is about a person, such as in surveillance case reports), will implement the Public Health Conceptual Data Model structure as appropriate, will include the ability to associate incoming data with appropriate existing data (e.g., a report of a disease in a person who had another condition previously reported), and will function so that data can be accessed by standards-based commercial products for reporting, statistical analysis, geographic mapping and automated outbreak detection algorithms. A more detailed discussion of how an integrated system can be implemented (e.g., through a data warehouse or one fully integrated software application) is contained in section II.2 describing the concept of the integrated system and in section III, "NEDSS implementation options for state programs."

D ~ Develop active data translation and exchange (integration broker) functionality.

The understanding of this element has been a challenge for many individuals involved in NEDSS development and implementation at the state level. As described in the program announcement, this element supports data translation, data import and export, queuing and messaging for the dynamic bi-directional interchange of data using the Extensible Mark-up Language (XML) to and from the integrated data repository. In particular, the integration broker function will be able to send reports within the health department and to other public health agencies including the CDC. Data exchange functionality will be deployed with the ability to rapidly develop ad hoc data exchange interfaces without programming. Additional information on this subject can be found in section IV.4, "Electronic data exchange."



E ~ Develop transportable business logic capability.

This element deals with the middle layer of a multi-layer information system. Data validation, business rules⁴ for data accumulation, data processing, workflow implementation, data coding and decoding, registry mapping, and case management capabilities can be developed in an application server around the data repository. The term “transportable” in this context refers to the fact that in an ideal, multi-layer system the same business rules (or middle layer) can be used with little modifications for different database management systems, or for different stages of data entry and analysis within the same system

F ~ Develop data reporting and visualization capability.

This element refers to selective data reporting according to user need-to-know, statistical analysis, Geographic Information Systems (GIS) use and other visualization, display and mapping functions. It is strongly recommended that this capability be implemented using available commercial off of the shelf (COTS) software solutions through industry standards for access to the data repository. Some examples of such solutions are Crystal Reports, SAS, SPSS, EPI Info, ArcView. These applications can access information in modern database systems using industry standards such as ODBC and JDBC.

G ~ Implement a shareable directory of public health personnel.

Select information about pertinent public health personnel within state health departments, and possibly selected local health jurisdictions, will be listed in a standards-based directory. The directory will be shareable, maintained at the state or local level, and could be combined with directories from other state and local health departments and from CDC to function as a directory of public health personnel. The directory will capture information about the roles and expertise of personnel for the use by public health communication and notification systems. Only selected information could be made accessible to authorized individuals, if desired. The directory of public health personnel will be used to guide the flow of information within and among public health agencies for emergent and non-emergent purposes. Directories will be maintained using an industry-standard for electronic directories called the Light Weight Directory Access Protocol (LDAP).

H ~ Implement a security system and appropriate security policies.

This element calls for the development of standards, operating procedures and infrastructure for the secure transmission, processing and storage of sensitive or critical data and the support of sensitive or critical systems. This will include the secure Internet exchange of information based on a secure Internet connection that can work in concert with the CDC’s Secure Data Network (SDN). Security policies need to be implemented, with authentication based on industry standard digital certificates of security, secure tokens, and other applicable means as identified; access and control of data via selective integrated repository authorization; an encryption engine and appropriate use of encrypted data; and access control through a firewall for data routing to appropriate programs and other organizations. A more detailed discussion of some of these security issues can be found in the section IV.4, “Electronic Data Exchange.”

III ~ NEDSS implementation options for state programs

As discussed above, NEDSS is simply the description of an architecture and its elements. While the architecture and the elements are well defined and standardized, their implementation can be managed in several different ways. This demonstrates the flexibility of the NEDSS architecture, which allows states to choose the options that fit best in their environments. Regardless what path of implementation is chosen, the resulting systems will have common definitions of terms, standard codes, and structures that will make them similar enough to each other to interact as if they were one system. At the same time, this high level of flexibility can cause confusion and uncertainty for those making decisions on how exactly NEDSS should be implemented in their agency.

In deciding how to implement the NEDSS architecture, there are some basic questions that a state program needs to address, such as

Does the state want to adopt a NEDSS model that integrates multiple surveillance systems into one software application, or a model that leaves the systems on separate databases and applications but permits combining and processing the information from these systems using shared tools?

Should the state develop its own software solution, or should the state wait for the new CDC-produced, NEDSS-compliant software under development?

How many systems should be targeted for inclusion in NEDSS?

This section discusses some of these issues and provides some guidance in this decision-making process.

1) Fully integrated system

A ~ Description

A fully integrated system is one in which the different functional components of the system share the same user interface, business rules, and database. The user will launch the application, start from a common starting point (e.g., a name search screen), move through a common process (e.g., identify an individual already existing in the database and open that individual's record), and enter or edit information on a patient using the same input screens. The information is then saved in one database.

A fully integrated system can also accommodate information specific to one or more conditions, with a common case record screen linked to disease-specific screens that become accessible only if the disease of interest is selected. For example, in a fully integrated system including tuberculosis and STDs, if the patient is identified as having TB, then the TB screen will become visible with all the information that the TB case investigator needs to process. In case of a patient with an STD, a different screen will show up with STD information. The basic case record information is stored in one database for both TB and STD patients.

LESSONS LEARNED: KANSAS SYSTEM (HAWK)

Q. What is the integration model your state chose (i.e., fully integrated, separate systems sharing common tools, etc.)?

A. We chose to create one integrated system for vaccine-preventable diseases, TB, and infectious diseases. We are planning to include HIV and STD's records through a data warehouse model.

Q. Why did you choose this model?

A. This system enables easy and rapid modifications that can affect multiple programs. We wanted to develop a system in line with NEDSS architectural elements that could serve the programs involved. We had been dissatisfied with the status of the software provided by CDC for TB and infectious disease surveillance (i.e., TIMS and NETSS). We wanted to take full advantage of the features, such as patient-based system, available only through a common integrated system.

Q. What were the resources required for developing the system?

A. The cost of contractual development was about \$300,000. One full FTE was funded for a system administrator.

Q. What are the barriers that you've had to face and overcome?

A. Funding was problematic and required a lot of work with different sources of grants. The development of the system (not necessarily the technical, but the conceptual component) required considerable time for program staff. Since the system was developed by an outside consultant, program staff needed to be vigilant and assure that the consulting firm understood how public health business works.

Local health department and state users were somewhat reluctant to drop their current systems and adopt a new one, even when the current system was clearly inadequate to meet their programmatic needs. Training users for the new system required a considerable investment of time and staff. The HAWK system was designed and developed using 1998 CIPHER standards. Those standards have evolved considerably and the Public Health Conceptual Data Model (PHCDM) project began later. Some changes have been made to keep HAWK compatible, but at added expense and time so that enhancing HAWK features for users has been delayed. Presently we are sending transmissions on reportable diseases using SDN, but CDC support for this has been uneven.

We would prefer to have all acute diseases reported and tracked using one system, but some specific CDC programs have been reluctant to embrace this concept and have hindered progress at the state and local level. Import and export capabilities have been weak in many CDC-produced software products.

Q. What would you do differently if you had to do it over again?

A. Allocate more resources and staff to development phase of both HAWK and NEDSS integration.

Work with laboratories and other partners more closely in earlier development of the system to allow easier electronic transfer of data.

Add mapping and graphing capabilities to system.

Submitted by: Gail Hansen and Gianfranco Pezzino



There are some advantages in having all the information from multiple surveillance programs included in one system. The first, and most obvious, is that the information from different programs will be easily accessible in a common, integrated fashion. That will allow users, for example, to ascertain if a patient diagnosed with disease A was also reported at another time as having disease B, or to easily prepare reports and charts that combine diseases from different programs. Another advantage is that having to support, maintain, and upgrade one system is usually easier and more efficient than doing the same across multiple systems. For example, if a new field needs to be added to the patient's information screen, once that screen is modified in the application it will be available to all the surveillance programs using the integrated system.

On the other hand, establishing such a system is usually a complex task that requires careful planning and considerable resources during the planning and development stage. The cost of developing such systems may be high, in the order of as much as several hundreds of thousands of dollars. In addition, security issues may be complex, since presumably staff belonging to one program should need to have only limited access to information related to other programs. Finally, it is often challenging to have "buy-in" for an integrated system project from the staff in programs that have traditionally worked independently from each other using separate software programs.

2) Separate systems sharing common tools ("data warehouse")

A ~ Description

A data warehouse is a database that obtains information from records stored in one or more other databases. Data warehouses are usually batch-updated periodically, and they can contain enormous amounts of data.

The concept of the data warehouse has been implemented for years by information technology managers who felt the need to separate basic database operations such as data entry, updates, and edits from data query and analysis. The underlying idea was that processor-intensive tasks like complex queries and analyses could be performed on a replica of the database, allowing users free access to the original database to perform actual changes of its records. In the NEDSS environment, the term data warehouse has become synonymous in most cases with the combination of information abstracted from multiple databases for analysis, queries, and display purposes.

The data in a data warehouse is typically historical and static, and may also contain numerous summaries. A data warehouse can be structured to support a variety of analyses, including elaborate queries on large amounts of data.

The concept of data warehouse is relatively simple to understand keeping in mind the multi-layer architecture of modern database systems. The third layer (i.e., the database) in each of these systems can be accessed through common tools (e.g., ODBC), and the combined information can be displayed or can be saved as a new database. For example, a SAS session can be started. SAS will connect to database A containing TB records and database B containing HIV records. Following the selection criteria decided by the user, SAS will then display together records from the two databases and will run queries on the combined information. A new database containing records from the two sources can also be created, although for many operations and procedures this step may not be necessary.

This process of linking information from different sources is relatively simple if the sources comply with the same standards for variable definition, coding, and grouping. The larger the difference between the sources, the more complex this process.

LESSONS LEARNED: MISSOURI'S INTEGRATED PUBLIC HEALTH SYSTEM

During 1992-93, the Missouri Department of Health (MDOH) developed an Information Strategy Plan to replace the over 60 stovepipe applications in the department with a single fully-integrated system. There were two major objectives. One was to enable MDOH and the state's local public health agencies to provide better service to their clients. By having access to all of the public health information about a client in one database, including any environmental threats to the individual's health, the public health agencies are better able to serve the entire health needs of the client. The second major objective was to enable the public health officials in the state to do better and more complete research across the many public health programs.

Although the Information Strategy Plan was completed in 1993, MDOH did not begin making significant progress on the development of the integrated system until 1996 due to a lack of funding. The system is expected to be completed in 2002 at a total cost of about \$25,000,000 (including development, equipment, and installation of the statewide network). During this period, the number of information systems staff in MDOH has grown from 35 to 92. In addition, MDOH is using seventeen contract consultants to assist with the development of the integrated system. MDOH has encountered several barriers to developing the integrated system, including:

~ Resistance to the system by MDOH staff. Many of the programs preferred their dedicated systems because they had more control over them and were skeptical of the department's ability to develop a single integrated system that would meet their needs. Success in implementing the early phases of the integrated system demonstrated both MDOH's ability to develop

the system and the many advantages of the integrated system over existing stovepipe systems, all but eliminating the resistance.

~ Funding. Most of the funding for public health functions is categorical. The sources of the categorical funding have been reluctant to provide funding to integrated systems.

~ Rapidly changing technology. During the past three years, portions of the system have had to be redeveloped to more effectively use new technology that has become available.

~ Complexity of a large integrated system. Changes created by one public health program can affect other program's data. Thus, coordinating the many requirements of the different programs in MDOH is a major task and programs need to be willing to sometimes accept less than the "ideal" system for their program to gain the benefits that the integrated system provides to all of public health.

From the beginning, the MDOH Directors have supported the development of the integrated system. Without this support, I do not believe an integrated system development would be successful. If I were doing it over, I would be sure that the first programs implemented on the system were managed by strong supporters of the integrated system. Also, I would have encouraged greater participation in the design by representatives of local public health agencies. Finally, if I were beginning today, I would initially design the system to run on a multi-tier architecture with a Web front end, technology that was still immature during the early stages of Missouri's development.

Submitted by: Garland Land



LESSONS LEARNED: NEW YORK STATE

Q. What is the integration model your state chose (i.e., fully integrated, separate systems sharing common tools, etc.)?

A. New York State (NYS) has selected an integration model that builds new systems, modifies existing systems, and builds NEDSS specific components in parallel. The goal is to integrate existing systems under the NEDSS umbrella while maintaining full functionality of existing systems. NYS continues to evaluate the overall best integration strategy to adopt. Currently NYS has adopted a unified development and deployment environment for all applications that will be built under the NEDSS umbrella. NYS has further decided to build a conceptual data model and logical data model as a step towards deciding on the type of integration that best meets present and future needs. The data model will bring together all four systems that are currently in use: the Clinical Laboratory Information System (CLIMS), Communicable Disease Surveillance System, Electronic Clinical Laboratory Reporting System (ECLRS), and the West Nile Virus Surveillance system.

Q. Why did you choose this model?

A. NYS has a significant base of operational systems utilizing the NYSDOH Health Information Network (an Internet based network). In planning the integration of these systems, all systems must remain fully functional during the integration process. NYS is initially looking at a middle tier data integration layer that would allow the existing systems to share information while maintaining functional status during the development and implementation of the new system.

Q. What were the resources required for developing the system?

A. The development of the Integrated Data Repository requires a team of about 15 people from Information Systems and the Division of Epidemiology. The initial project of building a Logical Data Model is expected to take 6 months. A full estimate of resources necessary will be developed as the planning and design process proceeds.

Q. What barriers did you have to face and overcome?

A. One significant barrier is the requirement to maintain and enhance the existing systems while devoting time and resources to the new system. Effective communications with all the stake holders is a continuing problem, since they are routinely busy in day-to-day surveillance activities, and getting their time is a major challenge. Finding personnel to work on the project continues to be a significant barrier.

Q. What would you do differently if you had to do it over again?

A. Adopting a development cycle that includes designing, prototyping, and implementing in many small steps. A previous project did a full design phase, built the system, and implemented it without an effective prototype. This led to much criticism and requests of changes at the time of implementation. We also recommend providing adequate time for each of the stages of the project, and not letting a preconceived schedule drive the project. Involvement of end users with adequate time to gather and review systems requirements and needs is essential to a successful project.

Submitted by Perry Smith and Michael Davesson



B ~ Advantages and disadvantages

One of the advantages of a data warehouse model is that it does not require the information from all the programs to be stored on one system. That means, for example, that the staff in the STD and HIV programs can continue to use the dedicated software that serves their needs. These software applications are often optimized to patient management procedures very specific to each program. Incorporating all those procedures into a single, integrated software application can be a complex process requiring considerable resources. Since the data warehouse lets staff continue to use their own software, it makes it easier to obtain “buy-in” from the programs involved in the process. In some circumstances, a data warehouse can be the easiest and most cost-efficient way to integrate information from different surveillance systems.

A data warehouse has several disadvantages. The approach followed by a data warehouse could be defined as “back end” integration, that is, integration after the information has been entered, validated, and processed. This type of integration does not provide relief from such problems as duplicate and redundant data entry or conflicting validation standards, problems that are better addressed by a fully integrated, “end-to-end,” system.

Another limitation that needs to be considered is that the sources of information where the records are originally stored need to be as similar as possible in their model and structure. For example, if databases A and B use the same variable name and format for AGE, records can be extracted from the two databases much more easily than if database A stores age in a text variable called AGE_AT_DIAGNOSIS and database B stores age in a numeric variable called AGE. The format of the physical files (e.g., Oracle, SAS, Microsoft SQL, Microsoft Access, etc.) is not as important as their models and data definitions. Also, since the records in the data warehouse must need to be updated periodically from their sources, and for this reason a data warehouse may not contain the most recent version of the information stored in the source databases.⁵ In general, a data warehouse is not a suitable solution component for creating new records or updating existing records: all editing functions need to happen in the source databases. Finally, a patient-based system (where individual information is entered only once, and multiple records can be linked to the same individual) may be difficult to implement in a data warehouse; most of the time, if the same individual has a record in database A and one in database B, these will appear as two individuals and two records when combined in the data warehouse.

Sometimes a combined approach including both an integrated system and a data warehouse system seems to work better, as in the case of Colorado (see the following box).

LESSONS LEARNED: COLORADO DEPARTMENT OF PUBLIC HEALTH AND ENVIRONMENT

The Colorado Department of Public Health and Environment is integrating its computer systems with a combination of top down and bottom up efforts. For the last two years, we have been building systems that integrate smaller areas of the department. The Colorado Electronic Disease Reporting System (CEDRS) integrates a number of separate disease reporting systems. The Integrated Registration and Information System (IRIS) brings together the client management functions of several categorical programs. The Laboratory Information Management System (LIMS) manages the data for separate laboratories and tests. The Colorado Health Information DataSet (COHID) allows local health departments to query different databases within the department from one interface. All of these systems have been built with a common set of tools: Microsoft Visual Studio, Integrated Information Server, SQL Server, and SAS for statistical reporting.

The department plans to use NEDSS to bring these smaller integration efforts together. We are currently modeling the data and processes involved in providing laboratory results to the disease surveillance functions. We are building a data model that conforms to the Public Health Conceptual Data Model, and contains the specific data elements that disease surveillance requires from the laboratory. However, we will design the model so that it can expand to incorporate other department functions as we go further. We will complete enough of the top level of the data model to ensure that it can become a department-wide model. Then we will drill down to enough detail to build a repository that contains the data elements that the laboratory

is making available. We will build this portion of the repository. LIMS will export the data and CEDRS will import it.

The department has chosen this approach to break the effort of integrating our systems into manageable pieces. The people who use these systems must see results to remain interested in the process. We have learned a great deal about system integration by building the four systems described above, and we have produced systems that are in use today. We could not have achieved that with a completely integrated effort. These systems serve as proofs-of-concept that system integration can work.

Our work on laboratory results has reinforced the decision to use a repository as an intermediate step. Disease surveillance does not want the myriad of codes that the laboratory produces. They want to distill these results into a smaller number of outcomes. Other laboratory customers may wish to transform the results in different ways. Also, it is not clear that we can import laboratory results directly into CEDRS without some manual role in deduplication. The repository allows the laboratory to publish results while its customer's work on the steps needed to import and use the data.

The biggest challenge is that a long-term, integrated effort must be accomplished with small, individually funded steps. Programmers may like system integration but customers want results. We have to balance both needs to succeed.

Submitted by: Richard Hoffman



3) CDC-developed NEDSS software

The NEDSS initiative was officially launched by CDC in the early part of 2000. The NEDSS initiative has been directed by input from state partners, as well as current principles of information system architecture. In the several years before NEDSS was launched, CDC staff became increasingly aware that the software produced by CDC programs that most states traditionally have used for many years could not fit in the architecture of a modern, integrated information system, and could not meet the needs of all the states involved. In particular, it became clear that:

- Most of the existing CDC-developed software was not fully compliant with the NEDSS specifications;
- The use of CDC-produced software could no longer be assumed to be the best solution for all state-based programs, and some states have implemented or are developing new NEDSS-compliant systems for their own use;
- For some state-based programs, CDC-produced software solutions remain an attractive option to implement NEDSS.

Based on interest expressed by states, CDC has initiated the development of a base system that will be offered to states wishing to use it as a resource to help them in the implementation of NEDSS. States will not be required to adopt the CDC base system if they prefer to develop other software applications that meet the NEDSS standards. The flexible NEDSS architecture will ultimately allow the adoption of mixed solutions that include some components produced by CDC and others developed locally. The CDC software will include multiple components:

A ~ Base system.

CDC has developed a contract with an experienced web software development company, Computer Sciences Corporation, to develop a NEDSS base system able to capture basic information for nationally notifiable disease surveillance. This will replace the old NETSS system for the purpose of reporting national notifiable diseases to CDC, but it will have much greater functionality that will enable it to be a platform for all eight NEDSS elements. The base system will also include a laboratory module capable of importing HL7 messages. This application is scheduled to be introduced for pilot testing in mid 2001.

B ~ Program-specific components.

It is expected that the base system will be joined by other program-specific modules. At least ten CDC program areas are currently planning NEDSS-compliant applications, and some are already developing their software. These modules will be designed to work with the data entry module in the base system. However, individual state requirements may affect how database tables will be maintained. At this point it is unclear when these applications will be completed.

The adoption of CDC-produced software may be the easiest way to implement NEDSS for states that have not already committed resources towards the development of local applications. While a final assessment will be possible only after the development of the new software is completed, it is likely that the CDC software will fully implement the NEDSS architecture through high-quality products. The CDC-developed systems are expected to be patient-based, meaning that an individual reported with multiple conditions should appear in the system as one individual with multiple records of events. However, the degree to which the different components will be integrated with each other is still unclear. For example, how easily will the user be able to access records stored in different components (for example, an STD and an HIV record) without running multiple

applications or using different data entry screens? This and other questions will probably be answered in the next few months. However, there is no doubt that the use of CDC-produced software in many cases will relieve states from the burden of having to pay for the cost of the development and support of local solutions.

4) State-developed applications

Several states have implemented or are developing NEDSS-compliant surveillance applications. Some states that chose this path are Colorado, Florida, Kansas, and New York. Some of these states have demo or training sites available for evaluation purposes. In general, these applications are custom-made and tailored to each state's program needs. However, the code is usually available to other states wishing to examine and use it.

States that decide to develop their own NEDSS software for surveillance purposes will probably end up with more flexible products that can be adjusted to match detailed, state-specific program activities and procedures. Programs for which no CDC-produced software is available can be easily integrated into a new, state-produced system. However, these states will need to mobilize the necessary resources, both in cash and in staff, and to plan for the ongoing support of their new software. They will also have to pay attention to issues of linkage between their software and that of the CDC (e.g., case definition), so that records can easily be sent electronically to the CDC meeting NEDSS and CDC program specifications. This linkage can be complex and tricky, and troubleshooting activities can be expected to be primarily the state's responsibility. Given the fact that NEDSS is meant to result in a national disease surveillance system, a good linkage between state-based and CDC systems should remain a top priority.

The flexibility of the NEDSS architecture definition is such that these two solutions (that is, adopting the CDC-produced software or developing new software locally) are not mutually exclusive. It will be possible for states to adopt certain components produced by CDC (e.g., the base system) and create additional "plug-in" modules that can be linked to those components. This can be a successful model that could utilize benefits from the CDC-produced software and retain the flexibility that comes with developing software locally.

5) How many systems should be targeted for inclusion in NEDSS?

As fits the general definition of a system architecture, NEDSS is not targeted to any specific surveillance system. This architecture has been defined broadly enough that it can be applicable to public health surveillance systems dealing with a variety of different conditions. However, for practical and funding reasons the first NEDSS implementation efforts have been targeted primarily to communicable disease surveillance systems. This approach will allow an incremental implementation strategy that in most cases can be more manageable than one that tries to integrate many diverse surveillance systems at once. It is expected that after the successful implementation of the first NEDSS-compliant products, NEDSS will soon be expanded to include surveillance systems in chronic disease, occupational and environmental health, maternal and child health, and other fields.

Most state-based people who have been involved in the NEDSS implementation efforts would agree that a state should not include more than a few (e.g., three or four) programs in the initial implementation. This is particularly true when the state decides to develop the software locally. At the same time, when planning the implementation of NEDSS, one should look ahead and identify as many systems as possible that are eventual candidates for inclusion in the new integrated system. Although the actual inclusion of all these systems may not happen for some years, this information will be important in the data modeling and system definition necessary before the new NEDSS system is implemented.



Some of the factors to consider in deciding what systems to include in an integrated NEDSS project are source of the data (e.g., systems that receive records from the same source could benefit from being integrated if the integration results in less burden on the reporting entities), programs served by the systems, users of the systems, similarity of the systems in composition and architecture, technical complexity of the integration process, and other factors such as data access and confidentiality policies. Tight integration of systems of information belonging to programs that have little in common in their objectives, source of information, target populations, or procedures is not going to be helpful, is not one of the NEDSS goals, and is not recommended. An integrated surveillance system should always be seen as a tool towards better public health practices, not as a goal in itself. The CDC has never indicated that they will require states to link specific surveillance systems electronically. The decisions regarding which surveillance systems should be included in NEDSS, and when, will be made by each state depending on local needs, resources, and priorities. For example, linking cancer registries and infectious notifiable diseases may be desirable in some states, but not in others. By defining a common architecture for those information systems, NEDSS makes these linkages possible, but it does not require them unless the state can see an advantage in doing so.

6) How to choose the best model

A ~ Do programs share goals, resources, and procedures to an extent that integrating the information would benefit their operations?

How similar are the goals and procedures of the programs? In some states, integrating TB and STD records into one software application may make sense and may facilitate reporting from Local Health Departments and other providers, if the two programs work closely and routinely share information with each other. In other states there may be little benefit in such an integration. Integration of systems belonging to programs with little in common makes little or no sense. In addition, programs with very different functions and procedures are more difficult to integrate into one system. The follow-up and case management for an HIV patient and an STD patient may be more similar than for a tuberculosis patient and a lead poisoning patient. The more similar the functions and procedures, the easier and more meaningful the integration.

B ~ Do the programs that could be integrated already use an information system application?

If so, how happy are the users with the features of the existing software? If, for example, your STD program staff already has a software program that performs the functions needed, it may be more difficult to convince the staff to switch to a new system shared with other programs. On the other hand, if the staff is dissatisfied with the current application, they may be more willing to consider switching to a new one. Also, how difficult would it be to move all functions of the existing application to a new, integrated system?

C ~ What resources are available (quantity and quality)?

Developing one integrated system requires the commitment of considerable resources. These resources include funds (ranging from a few hundred thousand to millions of dollars), the presence of skilled computer programmers (on staff or on contract), and time commitment from program managers to assure that the final product will meet their needs. An incremental approach is possible, but the system must be planned and designed from the beginning with consideration for the needs of all the programs that will eventually use it. When resources are scarce, a data warehouse model may be easier to implement in an incremental fashion.



On the other hand, a fully integrated system, once developed, usually requires minimal maintenance and support, and changes introduced into the system are applied automatically to all the programs included in the system. A data warehouse system may be more challenging to support, since a change in any of the data sources may require an adjustment in the programming code that extracts the information from that source.

D ~ What are the security requirements of each program?

If some programs have such strict security requirements that they would put a burden on the other programs to be included, this could be a deterrent to the adoption of one integrated system. While the NEDSS security standards meet all requirements of federal programs, the access restrictions may be so big that in practice they would defeat the purpose of having one integrated system.

IV ~ Fitting NEDSS into a state plan: resources and other issues

1) NEDSS requires state resources

NEDSS is a broad, multi-year initiative with the potential of dramatically improving public health surveillance in this country. To implement such an ambitious plan requires a considerable investment of resources. While federal funds available through the CDC are substantial, they will not provide all the resources that states need to fully implement this plan. Even the adoption of the CDC-produced software will require state programs to support some infrastructure. For example, Internet connectivity and Web site management are essential components of the NEDSS architecture, and they can only be developed at the state level. In fact, their scope is so much broader than the NEDSS project that they could be considered an essential capacity that every state public health agency must insure. Another important area requiring state capacity is database management, since many tasks will need to be performed on the NEDSS databases at the state and local level, regardless whether these databases are part of a CDC-developed or a state-developed solution. For the success of NEDSS, it is very important that decision-makers at the state level understand that information, its collection, and its quality are central to delivering quality public health programs, and it is not reasonable to expect that a “shrink-wrapped” CD-ROM from CDC will address all their information systems needs. Some state-based NEDSS solutions require more state resources than others, but no solution can be implemented without state resources; careful planning should take into account these requirements and make sure that the solution adopted can be adequately supported by the state programs involved.

In assessing the cost of the implementation of NEDSS, it is important to keep in mind that some of these expenses are actually already part of a state budget, under different categories. States routinely invest considerable amounts of money in information technology resources for individual programs or projects. The NEDSS model tries to integrate these efforts in a more efficient way, and in the long run it may actually save money by allowing the sharing of information technology resources and the replacement of only a portion of an existing information system, instead of the whole system as typically happens to proprietary or legacy applications (for example, a database could be upgraded while retaining most of the current data entry screen and middle layer application).

NEDSS is not a federal mandate, although it is possible that some future surveillance funding initiatives may be available only to states with NEDSS-compliant systems. In fact, NEDSS will relieve states from current mandates that require the use of CDC-produced software to receive some categorical funds. This will increase the states’ capacity to use more flexible and appropriate information systems.



Implementation of the NEDSS architecture for their public health surveillance information systems is in the best interest of each state. This architecture will provide a level of functionality, flexibility, and standardization that will enable state surveillance programs to interface with clinical providers more easily, to decrease the burden on these providers of reporting diseases, to obtain more accurate and complete information, and to use the information effectively.

2) The cost of NOT implementing NEDSS

While the implementation of NEDSS will require state-leveraged resources, states should notice that there is also a cost for those that decide not to adopt the NEDSS architecture. Part of this cost will be in the form of “missed opportunities” to improve the effectiveness and efficiency of public health programs, but in some cases there will be a direct negative fiscal impact. For example, states without a NEDSS-compliant information system will be unable to communicate electronically with other states and with CDC without implementing a complex system of data translators require substantial investment and increased work on part of the staff. Reporting requirements to CDC (often a requisite to receive categorical funds) will be more difficult and more expensive to fulfill without a NEDSS-compliant system. Electronic connections with health care providers and clinical laboratories will be more problematic in the absence of a standardized system architecture that can be used as a point of reference for those partners, and the information received from them will probably be less accurate. Some of the HIPPA requirements may force public health agencies to adopt electronic communication standards like NEDSS. Duplicate data entry into non NEDSS-compliant systems will divert precious staff time from other public health activities. Finally, only the implementation of NEDSS in all states will enable the establishment of a true national electronic disease surveillance system.

3) Connectivity and security issues

A ~ Internet connections

No matter how states decide to implement NEDSS, they need the ability to connect to the Internet. This connectivity needs to be governed by policies for authentication and security. States also need to be able to display data in both public and secure Web sites, and to access CDC’s public and secure sites, when authorized. It is worth noting that funds have been made available by CDC to many states through the bioterrorism preparedness initiative for the purpose of upgrading their connectivity capacity. The importance of assuring a robust connectivity network cannot be overemphasized. In the absence of such a network no integration or exchange of information outside each single agency would be possible.

B ~ Data Security

The issue of data security is very complex, and a comprehensive discussion is outside the scope of this document. Some key points need to be kept in mind:

1) Firewalls are important to protect information stored on a computer or a network that is connected to the Internet. Firewalls do not protect the information while it is being transmitted between two points or users on different networks. For example, a file stored on a computer placed behind a firewall will be protected from intruders (assuming that the firewall is properly configured, of course). If that file is sent as an E-mail attachment to a recipient that is not located behind the same firewall, that file becomes unprotected during the transmission, and if intercepted it can be accessed.

2) Information exchanged during a Web browser session is, by default, not secure, in the sense that the information is not encrypted and, if intercepted, can be easily accessed. This information can be made secure through the use of “digital certificates,” small files that allow

the browser to identify users and to encrypt all the information exchanged during a Web session. This technique is often referred to as Secure Socket Layer, or SSL. This is the technique adopted by the CDC Secure Data Network used, for example, by Epi X. It is also the technique used by many Internet commercial sites when transmitting confidential information such as credit card numbers. During an SSL session many common browsers display a notification that the session is being encrypted (e.g., a padlock icon on the browser screen).

3) Information exchanged through electronic mail is, by default, NOT secure. The use of SSL encryption techniques is not routinely applied to electronic mail. Some electronic mail applications (e.g., Lotus Notes) can encrypt messages, but the recipient must use the same application to decrypt and read the message. Electronic mail messages can also be encrypted using dedicated encryption software or a Virtual Private Network (VPN).

4) Information exchanged through an FTP session (e.g., file download) has security features similar to those of electronic mail. Unless encryption techniques are adopted by all parties involved in the FTP session, the information exchanged is not secure.

C ~ Rapid Communication

States must plan and implement rapid communication to appropriate authorities at the federal, state and local levels for both routine and urgent public health business and for distance learning. Such plans could involve electronic mail, telephone, fax, paging systems, broadcast fax, satellite, and other automatic or manual features.

4) Electronic data exchange

A ~ Standardization issues

The whole NEDSS project is built around the idea of enabling different surveillance information systems to communicate with each other electronically. One of the major barriers for that process is the lack of standardization in these systems. The process of standardization is complex, but it can be simplistically divided into the following components.

1) Data modeling

A data model specifies the relationship between the different elements in an information system, such as relationships between areas of contents (e.g., individual information, clinical records) or between variables. A Public Health Conceptual Data Model (PHCDM) has been developed by CDC and can be found at <http://www.cdc.gov/od/hissb/docs/phcdm.htm>.

2) Data definition

This includes a definition of the system variables (or fields). An example of data definition is that an individual be identified by a text field FIRST, length 25 characters, a text field MID_INITIAL, length 1, and a text field LAST, length 25. Systems using the same data definitions for their variables are much easier to connect electronically. The CIPHER group at CDC has produced data definitions for most variables included in the base system. These definitions are being re-evaluated and modified during the development of the CDC base system.

3) Coding

After a variable is defined, its content needs to be specified and standardized. A good example of the importance of using standardized codes is race and ethnicity. Two systems may have fields defined in the same way to capture race and ethnicity information (e.g., RACE, text, length 1, and ETHNICITY, numeric, length 2), but unless they use the same codes and groups



the information contained in those variables cannot be directly compared across the two systems. For laboratory reports, LOINC and SNOMED are two coding systems that standardize the description of tests and the results of most clinical laboratory tests. These two systems complement each other and are both part of the standards being developed for electronic laboratory reporting.

4) Messaging and transmission protocols.

Information systems that use the same definition and coding for their variables need to be connected and “talk” to each other. If the two systems are on the same network, as with two databases in the same agency, a variety of methods can be used to extract and combine records. For example, most modern database management systems use a protocol called ODBC that can establish electronic connections with other ODBC-compliant databases. Also, programming languages like SQL provide standard procedures to extract and manipulate records from different databases.

Exchanging information between systems not on the same network is more complicated. For example, when receiving electronic reports from laboratories. In these cases two things are needed: standardization of data elements and standardization of the protocols to transmit the information.

HL7 is one system used to standardize this transmission.⁶ An HL7 message is a standardized message that contains text characters. These characters describe the content of the message and the values of the fields contained in the message. For example, an HL7 message would include information indicating that starting at column 15 of the message there will be 25 characters with the last name of the patient. HL7 is becoming increasingly popular as an electronic data exchange tool in the health sector, including the transmission of laboratory test results, and most large clinical laboratories now have computerized information systems that can generate HL7 messages. One barrier to the expansion of HL7 is the scarcity of applications used in public health surveillance able to produce and read HL7 messages. HL7 messages can be transmitted through several methods, including telephone connections (i.e., modems), Internet connections, removable media (e.g., CD), and others.

XML is another messaging approach that is becoming more promising, and could become an effective and powerful tool to exchange information across different systems using Web browsers. Information can be embedded directly in Web browser pages, without the requirement to generate separate export files. XML and HL7 protocols are expected to converge in the next release of HL7 (ver. 3.0), which will include the capability to transmit HL7 messages using XML protocols.

B ~ The concept of “data routers”

Data routers can be defined as transfer stations for electronic records originating from and directed to multiple sites. An example of how a router can be used is the electronic transmission of laboratory reports. A system could be created that would allow reference national laboratories to send the results of their tests electronically to a central router. A software application on the router would decide the destination of each report, based on the information contained in the report label (e.g., patient’s or physician’s residence, type of test performed, etc.). At that point, the report would be sent to the appropriate state public health agency. Such a system does not currently exist, but has been proposed as a fast and efficient way to transmit laboratory results to state surveillance systems. The main advantage of this system would be easier development, support and maintenance, when compared to individual, state-based electronic transmission systems between these laborato-

ries and each state. For example, if a state changed some specification for the electronic data exchange (such as the Internet address of its database), there would have to be a change on the router application, and that would be completely transparent to the laboratories where the messages originate. Updates and changes in the software (such as the adoption of new versions of HL7) would also be easier to implement with a central router system.

One concern about such a system is confidentiality of the information being transmitted. A router-based system could be set up in such a way that the content of each message could be encrypted and accessible only by the addressee in the state of final destination. Only the information necessary for the router software to decide where the message should be forwarded would be available at the router level, and this information would not need to include any patient identifiers. This system could be compared to an overnight shipping process that requires a parcel sent from station A to station B to go through station C for routing, where nobody would be able to access the content of the parcel. At this point it is still unclear what legal barriers could be present in individual states to allow the adoption of such a system. Some states have legal requirements for laboratories to report certain test results directly or only to the state public health agency, and the legal authority to have those reports sent to other entities out of the state's jurisdiction could be challenged.

5) Legal and ethical issues

A comprehensive surveillance system architecture like NEDSS inevitably raises ethical questions and questions about legal authorities. Much attention has been devoted to the technical aspects of NEDSS, but as its implementation progresses these policy questions will likely become more and more relevant. Given that these matters touch the very heart of the public health system in this country, state and CDC officials should give them as much attention as the technical aspect of NEDSS. Here are two examples of issues that states and CDC need to address when considering how to implement NEDSS.

A ~ Authority to collect and exchange electronic disease records

Under the U.S. constitution, states have the authority to require health care providers to report medical information of public health relevance. State laws vary with regard to control of reportable disease data, and the NEDSS system needs to take into account individual state's responsibilities and authorities. There is no indication that NEDSS will be used by the CDC to obtain confidential information from state programs beyond what is already being shared. NEDSS allows, but does not require, the secure exchange of confidential information. What information will be shared, and with whom, are policy decisions that need to be addressed by those involved in that decision process.

Some state laws and regulations include specific instructions for the disease reporting process, including the mode of transmission (e.g., by mail or fax) and format of the reports (e.g., on forms supplied by the state agency). In these cases, the legality of electronic data exchange needs to be assessed, and laws and regulations may need to be updated.

In addition, some states require that certain information be reported to a local health department, and from there to the state department, or that laboratories report all positive results for notifiable conditions to the state agency of the state where the laboratory is located, regardless of the patient's residence. These and similar requirements need to be taken into account to make sure that they do not interfere with the ability to perform electronic disease surveillance.



NEDSS makes exchange of information among multiple database systems easier by establishing standards and protocols for those systems. However, NEDSS contains no specification of what information shall be shared and exchanged, and among whom. These policy decisions are not part of the NEDSS architecture, but the implementation of this architecture must take into account such issues. For example, when paper-based disease reports are mailed to individual state program directors it is clear that the content of those reports is shared only among those who generate and those who receive the reports, such as the TB nurse in a local health department and the TB state official in the state office. However, when the same reports are included into one integrated surveillance system with multiple users from multiple agencies working in multiple program areas, the question of who exactly should have access to the records becomes more problematic and needs to be discussed in detail during the early planning stage of such an integrated system.

There are states where laws and regulations establish who has authority to collect and access identifying information for certain notifiable conditions, and these requirements need to be taken into consideration when designing an integrated information system. For example, if state laws allow only local health departments to access identifying information on HIV-infected individuals, the authority of a state health agency to store that information on the state network could be challenged.

Technical solutions are available to allow users of an information system to access only the information that is relevant to their tasks. Careful planning and well-thought-out policies should guide the implementation of these solutions. While these issues are primarily the responsibility of individual states, model templates of laws, policies, and procedures may assist states in the development of their solutions.

Endnotes

¹ The term “business rules” in this context is not used in reference to general business procedures such as billing or contracting, but to information systems-specific procedures such as data validation.

² This is possible through several database access methods, such as ODBC, which stands for Open Database Connectivity, a database programming interface that provides a common language for applications to access databases on a network.

³ For a more detailed description of HL7, XML, LOINC, and SNOMED see below, “Electronic Data Exchange”.

⁴ The term “business rules” in this context is not used in reference to general business procedures such as billing or contracting, but to information systems-specific procedures such as data validation.

⁵ In some cases it may be possible to create “dynamic links” in the data warehouse application that will always display the most current information contained in the data sources.

⁶ HL7 is a very complex and powerful messaging system. It can be used not only across information systems that comply with the same standardized data definitions, but also across systems with different structures and formats, as long as they are able to generate and read an HL7 message.

FAQs ~ Frequently Asked Questions

Q. How long will states be required to use disease specific data systems such as HARS, TIMS, and STDMS? Will there be new versions of these programs soon? Will states be required to use them as a condition of grant support? How long will the existing programs be supported by CDC?

A. There will be new versions of these programs and they will be NEDSS compliant. The release of any new version of these programs will be at the end of 2001 at the earliest. States will not be required to use these programs as a condition of grant support, however data must be transmitted in a standardized format. The CDC will support the existing programs for at least another year since the new versions won't be ready before then.

Q. When will the CDC-supplied software to replace the Epi-Info-based NETSS software be available? Will states be required to change over? How long will the old software be supported? Will changing over be a condition of ELC or Emerging Infections Cooperative Agreement funding?

A. A Base System is planned to be ready for testing in October 2001. After testing has been completed, it is estimated that a production version will be available by early 2002. If states are doing their own development then their time frame for having full functionality is dependent upon their own resources. As far as we know, all states are planning to use a NEDSS compliant system. The old software will be supported for at least another year since the new system won't be ready in all locations before then. Changing to a NEDSS compliant system will be required for future CDC-funded surveillance activities.

Q. Does CDC intend for there to be data repositories serving several, many, or all states, where data from clinical laboratories will be processed and passed on to the states? Or will each state set up their own? Or will this be handled in software so it is transparent to states?

A. CDC is working to establish an infrastructure for electronic laboratory reporting. The project is still under development. Under this infrastructure, CDC will route lab results from national labs to participating states. CDC is providing this routing service because standards for point to point messaging through HL7 are not available universally. State programs will receive from CDC a standardized data translator that will read the HL7 files and transfer the records into their state integrated data repositories. The process will be fully automated. Messages will not be stored on the CDC router. In addition, these messages will be encrypted during the transmission. Participation in this system will be voluntary: each state will indicate which programs can be included, and where the electronic reports will be sent.

Q. What if a state IT office doesn't buy into the NEDSS architecture? Can CDC provide technical assistance to work with state IT staff?

A. As far as we know, all states are planning to use a NEDSS compliant system. Furthermore, our work with state health departments and chief technical officers has indicated support for the NEDSS effort.

Q. How will use of and access to the newly uniform NEDSS data passed by the states to CDC be regulated?

A. The NEDSS architecture provides states the ability to integrate efficiently and standardize the information contained in their multiple surveillance systems. It also allows states to transfer to CDC information they are willing and legally allowed to share. CDC will continue current data security policies and refine these as necessary.



Q. Will states report personal identifiers to CDC by using the new NEDSS base system?

A. There is no plan to include personal identifiers in routine transmissions of surveillance records to CDC.

Q. What are all these data types in the NEDSS Public Health Conceptual Data Model document that are neither case reports nor lab tests? How will states use these in epidemiology programs? Does it make sense to computerize these functions?

A. These data types are necessary to allow for an integrated, patient-centered system at the states. State activities include more than just infectious disease surveillance.

Q. What are the purposes of NEDSS? How will implementing it improve the public's health? What kinds of changes can we anticipate in our disease control programs as a result of implementing NEDSS?

A. NEDSS compliant systems will provide more timely and accurate information for faster and better decision-making and more rapid recognition of public health threats, allow for the integration of information systems for better analysis and for sharing of information, establish an electronic interface with laboratories and clinical care systems for more complete reports with less burden on data providers and initiate a standard architecture for surveillance that can be used for more than infectious diseases.

Q. How do the HIPAA privacy regulations constrain what NEDSS can do, if at all?

A. The HIPAA privacy regulation permits access to individually, identifiable health information for appropriate public health uses without further individual consent. NEDSS includes standards for security and encryption of these data that are HIPAA compliant. In addition, at the CDC level, NEDSS data will not include personal identifiers.

Q. How and when are states going to build injury surveillance, chronic disease surveillance, asthma surveillance, birth defects surveillance, cancer surveillance, and so on, using NEDSS?

A. NEDSS infrastructure and standards are supportive of surveillance outside of infectious diseases. When this NEDSS infrastructure is established in the different states, we anticipate this infrastructure will support other areas of public health.

Q. Is the laboratory reporting initiative of NAACR in collaboration with NPCR consistent with NEDSS?

A. Yes. CDC and our partners are working with NAACR to ensure that a uniform guide for messaging, consistent with NEDSS, is developed.

Q. Does NEDSS require states to combine all their surveillance records from different programs?

A. No, NEDSS encourages, but does not require, states to integrate as many information systems as they deem appropriate.



Glossary of Terms and List of Acronyms

Architecture (network): The design of a communications system, which includes the hardware, software, access methods and protocols used. It also defines the method of control; for example, whether computers can act independently or are controlled by other computers monitoring the network.

Architecture (computer): The design of a computer system. It sets the standard for all devices that connect to it and all the software that runs on it. It is based on the type of programs that will run (business, scientific) and the number of them run concurrently.

Architecture (software): The design of application or system software that incorporates protocols and interfaces for interacting with other programs and for future flexibility and expandability. A self-contained, stand-alone program would have program logic, but not a software architecture.

Base system: With regard to NEDSS, the Base System is a platform upon which many public health surveillance systems, processes, and data can be integrated in a secure environment.

CIPHER: Common Information for Public Health Electronic Reporting – a set of standards or guidelines for data representation and code values which include specifications for representing concepts as well as standard code lists for coded elements. The CDC and its partners in public health have designed and implemented information systems to support surveillance for specific diseases and adverse health conditions.

COTS: Commercial Off the Shelf – refers to ready-made merchandise that is available for sale.

Data Model: The product of the database design process which aims to identify and organize the required data logically, through a set of mathematical equations, and physically, through location within a central data warehouse.

Data Repository: A database of information about applications software that includes author, data elements, inputs, processes, outputs and interrelationships. A repository is used in an application development system in order to identify objects and business rules for reuse. An IDR, or integrated data repository, would be a linked or bridged set of data repositories.

Data Warehouse: a generic term for a system for storing, retrieving, and managing large amounts of any type of data from single or multiple sources; often includes sophisticated compression and hashing techniques for fast searches and advanced filtering. The terms relational, network, flat, and hierarchical all refer to the way it organizes information internally.

Firewall: Software and/or hardware that protects systems from access by unauthorized users and programs.

JAD: Joint Application Design – JAVA decompiler which are able to read one or more JAVA class files and converts them into JAVA source files which can be compiled again.

JAVA: An object oriented programming language developed by Sun Microsystems.

JavaScript: A scripting language widely used on the Web. JavaScript is embedded into many HTML pages.

JDBC: JAVA Database Connectivity – a standard that allows JAVA programs to interact with any SQL compliant database.

HIPAA: Health Insurance Portability and Accountability Act – the Administrative Simplification provisions of the HIPAA Act of 1996 are intended to reduce the costs and administrative burdens of health care by making possible the standardized, electronic transmission of many administrative and financial transactions that are currently carried out manually.



HL7: Health Level 7 – a series of standards for the messaging of clinical data.

HTML: Hyper Text Markup Language – document format used on the World Wide Web. Web pages are built with HTML tags, or codes, embedded in the text. HTML defines the page layout, fonts and graphic elements as well as the hypertext links to other documents on the Web. Each link contains the URL, or address, of a Web page residing on the same server or any server worldwide, hence "World Wide" Web.

HTTP: HyperText Transport Protocol – the communications protocol used to connect to servers on the World Wide Web; primarily functions to establish a connection with a Web server and transmit HTML pages to the client browser.

Information technology: The processing of information by computer.

LDAP: Lightweight Directory Access Protocol – a vendor-independent, open, network protocol standard. It is platform independent and supports interoperability in the same fashion as a Simple Mail Transport Protocol (SMTP).

LOINC: Logical Observation Identifiers, Names and Codes – a set of code standards that identifies clinical questions, variables, and reports. LOINC comprises a database of 15,000 variables with synonyms and cross-mappings; it covers a wide range of laboratory and clinical subject areas. The formal structure has six parts: component, property measured, time aspect, system, precision, and method.

NETSS: National Electronic Telecommunications System for Surveillance – a system by which each of the states and territories and two large cities in the United States transmit data to the Centers for Disease Control and Prevention (CDC) for weekly examination and publication.

ODBC: Open Data Base Connectivity – a standard database access method developed by the Microsoft Corporation.

PHCDM: Public Health Conceptual Data Model – a high level conceptual model, developed as part of the CDC NEDSS initiative, which provides the foundation for standardization of public health data collection, management, transmission, analysis, and dissemination.

Plug in: An auxiliary program that works with a major software package to enhance its capacity.

Programming Code: A language used to write instructions to the computer. It allows the programmer to express data processing in a symbolic manner without regard to machine specific details.

SDN: Secure Data Network – the CDC project to allow for secure data transfer between state and local health departments and the CDC across the Internet.

SENSOR: Sentinel Event Notification System for Occupational Risks – the underlying goal of SENSOR is the prevention of occupational disease and injury. As one of the major CDC/NIOSH surveillance programs, SENSOR promotes the more general goals for surveillance including: identifying new, unrecognized occupational diseases, injuries, and hazards; identifying sentinel diseases, injuries or hazards, the occurrence of which represents a failure of prevention; determining and tracking the magnitude and distribution of those diseases in question; disseminating information to aid the public and government in decision-making.

SNOMED: Systemized Nomenclature for Medicine – a nomenclature classification for indexing medical vocabulary, including signs, symptoms, diagnoses, and procedures; defines code standards in a variety of clinical areas, called coding axes. SNOMED can identify procedures and possible answers to clinical questions coded through LOINC.



SQL: Structured Query Language – a standard language for requesting information from a database.

SSL: Secure Socket Layer – a method for the encrypted transmission of data across TCP/IP.

TCP/IP: Transmission Control Protocol/Internet Protocol – standards that are the basis for data transmission on the internet, over LANs (local area networks), and WANs (wide area networks).

XML: Extensible Markup Language – a specification developed by the World Wide Web consortium. XML is designed especially for web documents, and it allows designers to create their own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations.

